

1 MORRISON & FOERSTER LLP  
2 MICHAEL A. JACOBS (Bar No. 111664)  
mjacobs@mofo.com  
3 KENNETH A. KUWAYTI (Bar No. 145384)  
kkuwayti@mofo.com  
4 MARC DAVID PETERS (Bar No. 211725)  
mdpeters@mofo.com  
5 DANIEL P. MUINO (Bar No. 209624)  
dmuino@mofo.com  
6 755 Page Mill Road, Palo Alto, CA 94304-1018  
Telephone: (650) 813-5600 / Facsimile: (650) 494-0792

7 BOIES, SCHILLER & FLEXNER LLP  
8 DAVID BOIES (Admitted *Pro Hac Vice*)  
dboies@bsfllp.com  
333 Main Street, Armonk, NY 10504  
9 Telephone: (914) 749-8200 / Facsimile: (914) 749-8300  
10 STEVEN C. HOLTZMAN (Bar No. 144177)  
sholtzman@bsfllp.com  
11 1999 Harrison St., Suite 900, Oakland, CA 94612  
Telephone: (510) 874-1000 / Facsimile: (510) 874-1460

12 ORACLE CORPORATION  
13 DORIAN DALEY (Bar No. 129049)  
dorian.daley@oracle.com  
14 DEBORAH K. MILLER (Bar No. 95527)  
deborah.miller@oracle.com  
15 MATTHEW M. SARBORARIA (Bar No. 211600)  
matthew.sarboraria@oracle.com  
16 500 Oracle Parkway, Redwood City, CA 94065  
Telephone: (650) 506-5200 / Facsimile: (650) 506-7114

17 *Attorneys for Plaintiff*  
18 ORACLE AMERICA, INC.

19 UNITED STATES DISTRICT COURT  
20 NORTHERN DISTRICT OF CALIFORNIA  
21 SAN FRANCISCO DIVISION

22 ORACLE AMERICA, INC.

Case No. CV 10-03561 WHA

23 Plaintiff,

**ORACLE'S APRIL 5, 2012 BRIEF  
REGARDING COPYRIGHT ISSUES**

24 v.

25 GOOGLE INC.

Dept.: Courtroom 8, 19th Floor  
Judge: Honorable William H. Alsup

26 Defendant.

27

28

1       **I. GOOGLE INCORRECTLY STATES THE LAW ON THE SIGNIFICANCE OF  
2       SELECTION, ARRANGEMENT, OR STRUCTURE**

3       The Court's questions on selection, arrangement, and structure have uncovered the  
4       extremity of Google's position. Google is urging the Court to adopt bright-line tests for which it  
5       has no supporting legal authority and that are directly contrary to Ninth Circuit law.

6       **A. The Structure Of A Computer Program Does Not Itself Have To Be A  
7       Computer Program To Be Copyrightable**

8       Google argues the structure of the Java APIs is not protectable because the structure is not  
9       a computer program:

10       The selection, arrangement and structure of the APIs present a different situation. At  
11       any level of abstraction above the actual implementation (i.e., the source code), the  
12       APIs *no longer meet the statutory definition of a computer program*.

13       (ECF No. 852 at 5-6.) Google cites no judicial authority for its position.

14       There is no reason to require that the selection, arrangement and structure of a  
15       copyrightable computer program itself be a computer program. The structure of a motion picture  
16       does not need to fit the definition of "motion pictures" (17 U.S.C. § 101) to be copyrightable.  
17       Nor does the structure of a book need to be a book to be copyrightable. *See, e.g., Stewart v.*  
18       *Abend*, 495 U.S. 207, 238 (1990) (movie "Rear Window" infringed book copyright by copying its  
19       "unique setting, characters, plot, and sequence of events").

20       In the Ninth Circuit the test for determining the copyrightability of the selection,  
21       arrangement or structure of a computer program is set forth in *Johnson Controls*:

22       Whether the nonliteral components of a program, including the structure, sequence  
23       and organization and user interface, are protected depends on whether, on the  
24       particular facts of each case, the component in question qualifies as an expression of  
25       an idea, or an idea itself.

26       *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989). The  
27       test does not require non-literal components of a computer program to be a program to be  
28       protected.

29       Google gives *Johnson Controls* a tepid endorsement, but *Johnson Controls* is hardly  
30       alone. It has been settled law for more than 20 years that the non-literal components of computer  
31       programs qualify for copyright protection. *See, e.g., Computer Assocs. Int'l, Inc. v. Altai*, 982

1 F.2d 693, 702-03 (2d Cir. 1992) (“We have no reservation in joining the company of those courts  
 2 that have already ascribed to this logic.”); *Gen. Universal Sys., Inc. v. Lee*, 379 F.3d 131, 142 (5th  
 3 Cir. 2004) (“settled” that copyright protection extends to “a program’s nonliteral elements,  
 4 including its structure, sequence, organization, user interface, screen displays, and menu  
 5 structures.”). Google’s proposal, that the test should be whether the structure “divorced from  
 6 implementing code” qualifies as a computer program, would gut that law. (ECF No. 852 at 6.)

7 Google copied the detailed selection, arrangement, and structure of Oracle’s computer  
 8 programs, which are described in Oracle’s specifications and source code. There is no dispute  
 9 that the Android specifications and libraries have this same selection, arrangement, and structure.  
 10 Google conceded at the hearing that they are “the same.” (3/28 Hr’g Tr. at 49:23-50:1; *see also*  
 11 ECF No. 778 at 3 (“substantially similar”)). The specifications are documents, the class libraries  
 12 are source code programs. They are both protected as literary works, and they fit hand in glove.

13 To try to escape liability under *Johnson Controls*, Google argues that APIs are  
 14 indispensable (ECF No. 852 at 8), and that there is “no evidence” the APIs are creative (*id.* at 8  
 15 n.7). Google is wrong. The APIs are a detailed, intricate blueprint that is the product of over a  
 16 decade of development work, and the evidence will clearly show this. Indeed, one of the people  
 17 who worked on and “wrote the APIs for many class libraries at Sun” (Bloch Dep. 227:2-7),  
 18 including several of the APIs in suit (*id.* at 24:7-16), has many times publicly given a presentation  
 19 that extols both the creativity and the design *decisions* involved in writing APIs. *See*  
 20 <http://www.infoq.com/presentations/effective-api-design>. The APIs at issue are far more creative  
 21 than the pieces of source code Google hired contractors to write over a period of months, when  
 22 Google re-implemented the Java APIs according to the design it copied. The APIs easily meet  
 23 the threshold of creativity applied by the Ninth Circuit in *Johnson Controls*, which found that  
 24 “some discretion and opportunity for creativity exist in the structure” was sufficient to uphold the  
 25 district court’s preliminary injunction finding. 886 F.2d at 1176. Google has already conceded  
 26 that the 37 APIs meet the originality standard of *Feist*, so “[t]he jury therefore need not be asked  
 27 to address whether the APIs are original.” (ECF No. 823 at 9.) Google cannot retract its  
 28 admission now.

1                   Equally unavailing is Google's argument that the API structure is not copyrightable  
 2 because the API elements "can be arranged in whatever order a developer chooses, to bring about  
 3 whatever particular result that developer desires." (ECF No. 852 at 6.) How the developer  
 4 chooses to use the APIs does not determine the APIs' structure, nor could it determine designers'  
 5 selection of what elements to include in them. *See, e.g., Stern Elecs., Inc. v. Kaufman*, 669 F.2d  
 6 852, 855-56 (2d Cir. 1982) (rejecting challenge that video game was uncopyrightable because  
 7 player could affect the sequence of the game's sights and sounds).

8                   **B.       There Is No Blanket Exception To Copyrightability Of Expressive Selection,  
 9                   Arrangement Or Structure.**

10                  Google again argues that "Section 102(b) excludes protection for all systems and methods  
 11 of operation, without regard for whether they are original, creative, elegant, life-changing or  
 12 difficult to develop." (ECF No. 852 at 2.) Lacking authority for this extreme and unsupported  
 13 position, Google instead cites *Nichols v. Universal Pictures Corp.* This is ironic because, in  
 14 pressing the Court to adopt inflexible categories of uncopyrightability, Google cites the case in  
 15 which Learned Hand famously said of idea and expression: "Nobody has ever been able to fix  
 16 that boundary, and nobody ever can." 45 F.2d 119, 121 (2d Cir. 1930). The facts of that case are  
 17 stunningly different than this one: in *Nichols*, the only thing in common in the story of the two  
 18 plays was "a quarrel between a Jewish and an Irish father, the marriage of their children, the birth  
 19 of grandchildren and a reconciliation." *Id.* at 122. Here, Google didn't only adopt the ideas  
 20 underlying the 37 Java APIs, it painstakingly copied element by element, relationship by  
 21 relationship, until it had replicated several thousand elements from them. The case *ATC*  
 22 *Distribution Group, Inc. v. Whatever It Takes Transmissions & Parts, Inc.*, 402 F.3d 700, 705  
 23 (6th Cir. 2005), is about a transmission parts catalogue, and is just as far removed. But it  
 24 contradicts Google's interpretation of section 102(b). It states that Section 102(b) "codifies the  
 25 common law principle" that "protection is given only to the expression of the idea—not the idea  
 26 itself." *Id.* at 707 (citation omitted). It thus aligns with *Johnson Controls* and virtually every  
 27 other Circuit that has considered this issue. (*See* ECF No. 853 at 8 n.1.)

28                  Although Google asks the Court to unconditionally deny protection to even the expressive

1 elements of a work in every category listed in section 102(b), it has no principled way of defining  
 2 them. Google cannot decide whether the APIs are a “method of operation” (as it originally  
 3 claimed) or a “system” (as it now asserts), so it argues they are both. It claims the APIs are *either*  
 4 a system or a method of operation because “[p]rogrammers cannot use the APIs without relying  
 5 on the structure, sequence and organization of the APIs.” (ECF No. 852 at 8.) But no computer  
 6 program can be used without relying on its organization and structure, and still programs are  
 7 copyrightable. Indeed, Google’s immediately preceding argument that the APIs “can be arranged  
 8 in whatever order the developer chooses” (*id.* at 6) is contradictory. No program could survive  
 9 Google’s “test.”

10 **II. DISCUSSION OF CONTU**

11 Google tries to minimize CONTU. But Oracle and Google agree on one key point from  
 12 the CONTU report: “Should a line need to be drawn to exclude certain manifestations of  
 13 programs from copyright, that line should be drawn on a case-by-case basis by the institution  
 14 designed to make fine distinctions—the federal judiciary.” (ECF No. 852 at 4 (citing Final Report  
 15 of the National Commission on New Technological Uses of Copyrighted Works 22-23 (1979)).

16 It is surprising that Google would cite this portion of the CONTU report given that  
 17 elsewhere in its brief, Google advocates a bright-line rule that no API should be afforded  
 18 copyright protection under § 102(b) as a “system” or “method of operation.” Notably, two former  
 19 CONTU commissioners have stated that Google’s bright-line rule is erroneous. *See* 1-2 Nimmer  
 20 on Copyright § 2.03[D] (“It would, then, be a misreading of Section 102(b) to interpret it to deny  
 21 copyright protection to ‘the expression’ of a work, even if that work happens to consist of an  
 22 ‘idea, procedure, process, etc.’”); Arthur R. Miller, *Copyright Protection For Computer  
 23 Programs, Databases, And Computer-Generated Works: Is Anything New Since CONTU?*, 106  
 24 Harv. L. Rev. 977, 1036 (1993) (“Courts should continue to resolve questions of the  
 25 copyrightability and scope of protection of computer programs by using the flexible principles  
 26 that apply to literary works and not resort to arbitrary exclusion of particular software  
 27 elements.”). Their approach is consistent with the CONTU report’s emphasis on protectability of  
 28 expression, *see* CONTU Report at 19, and the earlier House Report’s statement that “Section

1 102(b) in no way enlarges or contracts the scope of copyright protection under the present law.”

2 H.R. Rep. No. 94-1476 at 57 (1976); S. Rep. No. 94-473 at 54 (1975).

3 In support of its bright-line approach, Google directs the Court to *Interfaces on Trial 2.0*.  
 4 Its central thesis is that copyright protection comes at the expense of interoperability, and harms  
 5 competition and the progress of science. But the level of copyright protection for interfaces is not  
 6 one-size-fits-all. Moreover, having intentionally fragmented the Java platform, Google cannot  
 7 wear the mantle of interoperability. Sun’s and Oracle’s API specification license relies on  
 8 copyright protection to *ensure* interoperability. The best way to promote the progress of science  
 9 is by encouraging investment in software development through strong copyright protection.

### 10 III. MERGER DISCUSSION

11 Google argues that even if the selection, arrangement and structure of the Java APIs is  
 12 expressive, “there is no way to ‘express’ the idea of the 37 APIs without repeating that selection,  
 13 arrangement and structure.” (ECF No. 852 at 10.) Google never identifies what it thinks the idea  
 14 of the APIs is, or could be, that would require them to be expressed in precisely this way, instead  
 15 of the millions of other possible combinations. It certainly would be difficult to articulate.

16 This is not a close call. Google is unable to cite any case with even remotely comparable  
 17 facts that holds its way. *Allen* was a case about a copyright on the rules for an educational game,  
 18 where the court found that merger applied because “[t]o hold otherwise would give Allen a  
 19 monopoly on such commonplace ideas as a simple rule on how youngsters should play their  
 20 games.” *Allen v. Academic Games League of Am.*, 89 F.3d 614, 618 (9th Cir. 1996).

21 Many courts have rejected Google’s approach to the merger doctrine. In *Control Data*  
 22 *Sys., Inc. v. Infoware, Inc.*, for example, the court rejected application of the merger doctrine to  
 23 components such as input/output formats, file layouts, and commands for the NOS operating  
 24 system, finding the appropriate analysis “is to compare the idea of NOS, that is the idea of  
 25 making an operating system for the Cyber computer, with the expression of that idea as embodied  
 26 in NOS.” 903 F. Supp. 1316, 1321-23 (D. Minn. 1995); *see also CMAX/Cleveland, Inc. v. UCR,*  
 27 *Inc.*, 804 F. Supp. 337, 354 (M.D. Ga. 1992) (“Defendants contend that the programs are only

28

1 similar in their file names and the sequence of the field names within each file. The selection and  
 2 arrangement of the field definitions within the files, however, are the expression of an idea.”).

3 This same approach was followed in the order denying the petition for rehearing in  
 4 *Engineering Dynamics*. Oracle’s citation to the appellate opinion was entirely consistent with the  
 5 petition order. It described the court’s emphasis on expression and did not claim the decision was  
 6 based only on how they “could have been structured in numerous ways.” (Compare ECF No. 833  
 7 at 2 with *Eng’g Dynamics, Inc. v. Structural Software, Inc.*, 46 F.3d 408, 409 (5th Cir. 1995).)

8 It is Google that mischaracterizes the court’s opinion. Google argues the selection  
 9 arrangement and structure reflect “judgments about what APIs are most useful, what options  
 10 should be made available to developers, and how best to organize the APIs” and that “[t]hese  
 11 choices were precisely the type of ‘functional judgments’ the Fifth Circuit explained *cannot* be  
 12 protected by copyright.” (ECF No. 852 at 10.) The *reverse* is true. These are the kinds of  
 13 judgments the Fifth Circuit held *are* protected by copyright. The court used the term “functional  
 14 judgment” to refer to decisions constrained by industry standards or science. It used this  
 15 language in confirming its earlier opinion remanding to the lower court to determine “whether  
 16 EDI exercised any judgment in formulating the input cards *or merely reflected the industry*  
 17 *standards and laws of engineering.*” *Eng’g Dynamics*, 463 F.3d at 410 (emphasis added).

18 The decisions made with respect to the API design were not governed by these factors.  
 19 The API designers had almost complete liberty in deciding what to include, evidenced by the  
 20 growth of the API packages from 8 in the initial release to 166 in 2004 (see ECF No. 852 at 2 n.3,  
 21 5). They were also free to structure the design how they wanted, because the structure of the API  
 22 design is for the benefit of the developers, and is not functionally required by the machine.  
 23 (Mitchell Opp. Rep., ECF No. 397-1, ¶ 23.)

24 Copyright law does not require that the API designs were chosen for an “aesthetic or  
 25 stylistic purpose” as Google contends, though many of them were. Choices made to “enhance the  
 26 capabilities and usability” would be quite enough. (See ECF No. 852 at 10.)

27 But even if Google could prove merger applied, it would still be liable for its “nearly  
 28 identical copying.” *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1444 (9th Cir. 1994).

#### **IV. THE JAVA APIs ARE INTEGRAL TO THE JAVA PLATFORM BUT ARE DISTINCT FROM THE JAVA LANGUAGE**

Oracle agrees that a small number of classes are necessary for the language. The overwhelming majority are not.

Google at least starts its analysis in the right place: *The Java Language Specification* (“JLS”). (The Third Edition is available at <http://docs.oracle.com/javase/specs/jls/se5.0/jls3.pdf>). That book describes the specification for the Java language. Google miscites from that book, however, in claiming that several classes in the `java.lang` package “should be treated as part of the language for purposes of Oracle’s copyright claims.” Google fails to mention that some of the classes to which Google refers are not actually provided in the JLS, and are not part of the language. *See JLS* (3d ed. 2005) at 6 (“The language definition constrains the behavior of these classes and interfaces, but *this document does not provide a complete specification for them*. *The reader is referred to other parts of the Java platform specification for such detailed API specifications.*”) (emphasis added). For example, the API specifications list over two dozen methods for class `ClassLoader`, but the JLS only requires one of them: “`defineClass`.” (Compare *id.* at 312 with <http://docs.oracle.com/javase/1.5.0/docs/api/java/lang/ClassLoader.html>).

Google also cites to a different book, without identifying it to the Court. That book is *The Java Application Programming Interface, Vol. 1*, the first of a two-volume set. The API specifications have been updated with each new Java platform release. While the Java language has stayed largely the same over time, the APIs have grown so large that the book set is no longer published in hardcopy book form, but is published online. What little the Java language requires of the Java libraries is spelled out in the JLS. The expression that Google copied is ***not there***. Instead, it is found in Oracle’s API specifications.

The rest of Google’s argument is a house of cards. Google argues all of java.io, java.util, and java.net are “fundamental” or “integral” to the language. (ECF No. 852 at 12-13.) Google’s evidence is the advertising copy on the back cover of the API book. (Trial Ex. 980.) On this issue, however, the parties’ experts agree that Google is wrong. Google’s expert testified that he agreed with the opinion of Oracle’s expert that “it was not necessary to include any particular

1 class or package (beyond perhaps a very few classes like Object and Class that are tied closely to  
 2 the Java language) for the Java language to function,” stating: “I think this is a reasonable  
 3 characterization of what’s needed for the language to work.” (Astrachan Dep. 230:19-231:16.)

4 Google claims that 15 additional packages are fundamental merely because they have  
 5 names *similar* to the four above. (See ECF No. 852 at 12.) But none of them *even existed* when  
 6 the language debuted, so cannot be said to be “fundamental” to the language. (Trial Ex. 980 at  
 7 xix.) For example, java.nio was not added to the platform until version 1.4 in 2002.

8 For the remaining 18 APIs, the most Google claims is that they “relate to features that are  
 9 standard in modern applications.” ECF No. 852 at 12. This does not make them integral to the  
 10 language and it does not excuse Google’s copying.

11 **V. COPYRIGHTABILITY OF COMPUTER LANGUAGES**

12 Because Google admits that the Java language and the class libraries are distinct (Google  
 13 Amended Counterclaims ¶¶ 1-3, ECF No. 51 at 13, 14), and its use of the Java language is not at  
 14 issue in this case, whether a language is copyrightable or patentable does not decide the case.

15 In any event, Google has not cited any federal judicial decision holding that a computer  
 16 language is not copyrightable or patentable. It does cite and quote from the opinion of Advocate  
 17 General Bot of the European Court of Justice in the *SAS Institute Inc. v. World Programming Ltd.*  
 18 case, that languages are not copyrightable under the European Directive. *See*  
 19 <http://curia.europa.eu/juris/celex.jsf?celex=62010CC0406&lang1=en>. Nevertheless, the  
 20 Advocate General also stated that program interfaces *could* be copyrighted “provided that they  
 21 contain some of the elements which are the expression of the intellectual creation of the author of  
 22 the work.” *Id.* at ¶ 60; *see also id.* at ¶ 81 (“As I see it, Directive 91/250 does not exclude  
 23 interfaces from copyright protection. It merely states, in the thirteenth recital, that ideas and  
 24 principles which underlie the various elements of a program, including those which underlie its  
 25 interfaces, are not protected by copyright under the directive.”). Thus the relevant portion of  
 26 Advocate General Bot’s opinion contradicts Google’s argument in this case.

27 Google also writes: “The fact that a programming language is not patentable subject  
 28 matter thus further supports the conclusion that it is on the ‘idea’ side of the idea/expression

1 dichotomy.” (ECF No. 852 at 17.) Even if that “fact” were true, Google has it backwards.  
 2 Because a computer language can be original, text-based, and capable of fixation—all qualities of  
 3 copyright—the better conclusion is that it is on the “expression” side of the dichotomy. This  
 4 conclusion is supported by the increasing popularity of coining new languages for motion pictures  
 5 and television shows, for which a language’s “sound” and “audience appeal” are considerations,  
 6 such as recently invented languages like Na’vi (from the film *Avatar*) and Dothraki (from the  
 7 HBO series *Game of Thrones*).<sup>1</sup> Why should years of creative work developing an original,  
 8 never-before-written language not be protectable?

9 **VI. THE APIs ARE COPYRIGHTABLE WHETHER OR NOT THEY ARE  
 10 PATENTABLE**

11 Google’s argument that Oracle may obtain patent protection but not copyright protection  
 12 for its APIs was directly contradicted by the Supreme Court in *Mazer v. Stein*: “Neither the  
 13 Copyright Statute nor any other says that because a thing is patentable it may not be copyrighted.”  
 14 347 U.S. 201, 217 (1954). The Federal Circuit held that software is eligible for both patent and  
 15 copyright. *Atari Games Corp. v. Nintendo of Am. Inc.*, 975 F.2d 832, 839-40 (Fed Cir. 1992).

16 Allowing copyright protection for computer interfaces makes sense because original  
 17 expressions in software are innovations of an incremental sort that Congress meant to encourage.  
 18 Trade secrecy law cannot achieve this goal because interfaces can be reverse-engineered. Patent  
 19 law, because of its novelty and nonobviousness requirements and examination process, protects  
 20 those substantial innovations, claimed as broadly and generically as possible, and in return gives  
 21 strong protection against even those who independently develop the same technology. Copyright  
 22 law protects innovations at a much finer level of detail (where original expression can be found)  
 23 than patents ever could, but only offers protection against the copyist.

24 **VII. GOOGLE INFRINGES ORACLE’S API IMPLEMENTATIONS**

25 For clarity, Oracle again states that, in addition to the copying of the selection,  
 26 arrangement, and structure of the APIs fixed in the specifications and the source and object code

---

27 <sup>1</sup> See, e.g., <http://dailytrojan.com/2009/12/02/usc-professor-gives-avatar-aliens-a-voice/>;  
 28 <http://www.nytimes.com/2011/12/12/arts/television/in-game-of-thrones-a-language-to-make-the-world-feel-real.html>.

1 implementation of the class libraries, Oracle is claiming copyright infringement based on: (1) the  
 2 thousands of elements Google copied from the Java API specifications into Android source code  
 3 and documentation; (2) the selection and arrangement of the names of the API elements Google  
 4 copied—an issue the Court expressly left open on summary judgment (*see* ECF No. 433 at 8);  
 5 and (3) Google’s creation of derivative works from the English-language descriptions of the  
 6 elements in the API specifications.

7 With respect to the literally copied code files, Oracle previously identified the 11 Java  
 8 source code files from which Google copied (ECF No. 780 at 3); of those, three files (including  
 9 `Arrays.java`) are part of Oracle’s source code implementation of the 37 APIs at issue. Twelve  
 10 Android source code files contain material copied from the 11 files; of those, two Android files  
 11 (`TimSort.java` and `ComparableTimSort.java`) are part of Google’s source code implementation of  
 12 the 37 APIs at issue. Both of these files include the `rangeCheck()` method that Google copied  
 13 from Oracle’s `Arrays.java`. Google also copied the comments to the `rangeCheck()` method.

14 But the real question is this: ***why is Google still distributing copied code?*** Just one month  
 15 ago, Google represented to the Court that “the allegedly copied portions of these files have all  
 16 been removed from Android” and asked the Court to find Oracle’s infringement claim ***moot*** on  
 17 that basis. (ECF No. 778 at 13.) Now Google admits that its representation is ***not true***. (ECF  
 18 No. 852 at 5 n.5) Google admits that code copied from Oracle is ***still available*** from its online  
 19 source code repository, yet acts as if it—with some of the best software engineering talent in the  
 20 world—is powerless to control its own repository. And Google does not explain how the copied  
 21 material “reappeared” in Froyo after having supposedly been removed “over a dozen version  
 22 releases ago.” (*Id.*) What is to keep it from reappearing again?

23 Google admits it is still publishing and distributing copied `rangeCheck` code, and that  
 24 copied code is still present in every phone made with every version of Android except Ice Cream  
 25 Sandwich. And that is a ***huge*** number: ***97.1%*** of Android phones active in the last 14 days have  
 26 Oracle’s `rangeCheck` in them. *See* <http://developer.android.com/resources/dashboard/platform-versions.html> (last visited Apr. 3, 2012). An injunction is needed to stop Google from continuing  
 27 to publish and distribute ***both*** copied code and copied APIs to Android phone manufacturers.

1 Dated: April 5, 2012

2 MORRISON & FOERSTER LLP

3 By: /s/ Michael A. Jacobs

4 *Attorneys for Plaintiff*  
ORACLE AMERICA, INC.

5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28